

CLAIM LISTING

Please amend claims 1, 18, 24, 34, 45, 52, 62, 79, 85, 95, 106, 113, 123, 129, 136 as indicated below.

1. (Currently amended) A method in a data processing system including a computer, comprising the steps of:

providing the computer with a software development tool having a user interface that is operable by a user to automatically reflect a modification in [[the]] source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an identification of a data structure with an attribute field in a database of data structures useable to form an object-oriented element from the data structure;

determining via the computer whether the data structure is associated with the source code;

when a determination is made that the data structure is associated with the source code, storing the identification of the data structure in a comment of the source code to relate the data structure with the source code and another determination is made via the computer as to whether the attribute field of the data structure is associated with an attribute in the source code;

when an alternative determination is made that the attribute field is not associated with an attribute in the source code, generating via the computer a source code modification, wherein the source code modification includes a new attribute in the source code [[from]] based upon the attribute field;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation

of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

2. (Previously presented) The method of claim 1, further comprising having the computer perform the following steps:

when it is determined that the data structure is not associated with source code, retrieving a portion of the data structure; and generating the source code from the portion of the data structure.

3. (Previously presented) The method of claim 1, further comprising having the computer perform the following steps:

when it is determined that the data structure is associated with source code, determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

when it is determined that a second attribute in the source code is not associated with a second attribute field in the data structure, removing the second attribute from the source code.

4. (Previously presented) The method of claim 3, wherein the step of removing the

SSK/slC

second attribute from the source code comprises the step of removing via the computer a method associated with the second attribute from the source code.

5. - 6. (Cancelled)

7. (Previously presented) The method of claim 4, further comprising the step of modifying via the computer the graphical representation of the source code to reflect the removal of the second attribute.

8. (Previously presented) The method of claim 1, wherein the step of determining whether the data structure is associated with the source code comprises the step of searching via the computer a comment in the source code for the identification of the data structure.

9. (Previously presented) The method of claim 1, wherein the step of determining whether the data structure is associated with the source code comprises the step of comparing via the computer a name for the source code with the identification of the data structure.

10. (Previously presented) The method of claim 1, further comprising having the computer perform the following steps:

retrieving access information for the database; and retrieving a portion of the data structure from the database using the access information.

11. (Previously presented) The method of claim 10, wherein the step of retrieving the access information comprises the step of retrieving via the computer the identification of the data structure and the access information from a configuration file;

12. (Previously presented) The method of claim 10, wherein the step of retrieving the

SSK/slc

access information comprises the step of retrieving via the computer the identification of the data structure and the access information from a comment of the source code.

13. (Original) The method of claim 10, wherein the portion of the data structure comprises the attribute field of the data structure.

14. (Original) The method of claim 1, wherein the source code comprises a class.

15. (Original) The method of claim 1, wherein the source code comprises a distributed computing component.

16. (Original) The method of claim 15, wherein the distributed computing component is an Enterprise JavaBean.[[TM.]]

17. (Previously presented) The method of claim 1, wherein the step of generating the new attribute in the source code comprises the step of generating via the computer a method in the source code to access the attribute field of the data structure.

18. (Currently amended) A method in a data processing system including a computer having source code that corresponds to a data structure within a database of data structures useable to form an object-oriented element from the data structure on a secondary storage device, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication to update the source code;

SSK/slC

determining via the computer whether the data structure is associated with the source code;

when a determination is made that the data structure is associated with the source code, storing the identification of the data structure in a comment of the source code to relate the data structure with the source code and another determination is made determining via the computer as to whether a first attribute in the source code is associated with a first attribute field in the data structure;

when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, generating via the computer a source code modification, wherein the source code modification includes removing the first attribute from the source code;

determining via the computer whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that [[a]] the second attribute field is not associated with [[a]] the second attribute in the source code, generating via the computer another source code modification, wherein the another source code modification includes generating the second attribute in the source code [[from]] based upon the second attribute field and associating the second attribute with the second attribute field;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification and the another source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source

code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

19. (Previously presented) The method of claim 18, further comprising the step of retrieving via the computer an identification of the data structure and access information for the secondary storage from a comment in the source code.

20. (Previously presented) The method of claim 19, further comprising the step of retrieving via the computer a portion of the data structure from the secondary storage device using the access information.

21. (Original) The method of claim 20, wherein the portion comprises the first and the second attribute fields.

22. (Previously presented) The method of claim 18, wherein the step of removing the first attribute from the source code comprises the step of removing via the computer a method associated with the first attribute from the source code.

23. (Previously presented) The method of claim 18, wherein the step of generating the second attribute in the source code comprises the step of generating via the computer a method in the source code to access the second attribute field of the data structure.

24. (Currently amended) A method in a data processing system including a computer having a memory device with source code and a secondary storage device with a data structure

SSK/slC

within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication that the data structure has been modified; and automatically reflecting the modification in the source code via the computer so as to avoid completely regenerating the source code;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification and the another source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

25. (Previously presented) The method of claim 24, wherein the step of automatically reflecting the modification comprises having the computer perform the following steps:

SSK/slc

determining whether a first attribute in the source code is associated with a first attribute field in the data structure; and when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code.

26. (Previously presented) The method of claim 25, wherein the step of removing the first attribute from the source code comprises the step of removing via the computer a first method associated with the first attribute in the source code.

27. (Previously presented) The method of claim 25, wherein the step of automatically reflecting the modification further comprises having the computer perform the following steps:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field in the data structure is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field.

28. (Previously presented) The method of claim 27, wherein the step of generating the second attribute in the source code comprises the step of generating via the computer a second method in the source code to access the second attribute field of the data structure.

29 - 30. (Cancelled)

31. (Original) The method of claim 24, wherein the source code comprises a class.

32. (Original) The method of claim 24, wherein the source code comprises a distributed computing component.

33. (Previously presented) The method of claim 32, wherein the distributed computing component is an Enterprise JavaBean.

34. (Currently amended) A method in a data processing system including a computer having a memory device with source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

determining via the computer whether the source code is associated with a data structure within a database of data structures useable to form an object-oriented element from the data structure;

when it is determined that the source code is associated with the data structure, storing the identification of the data structure in a comment of the source code to relate the data structure with the source code and determining via the computer whether a first attribute in the source code is associated with a first attribute field of the data structure;

when it is determined that the first attribute in the source code is not associated with the first attribute field in the data structure, generating via the computer the first attribute field in the data structure;

determining via the computer whether a second attribute field in the data structure is associated with a second attribute in the source code;

when it is determined that the second attribute field is not associated with the second attribute in the source code, removing via the computer the second attribute field from the data structure;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

35. (Previously presented) The method of claim 34, further comprising the step of when it is determined that the source code is not associated with the data structure, generating via the computer the data structure from the source code.

36. (Previously presented) The method of claim 34, wherein the step of determining whether the source code is associated with the data structure comprises the step of searching via the computer a comment in the source code for an identification of the data structure.

37. (Previously presented) The method of claim 34, wherein the step of determining whether the source code is associated with the data structure comprises the step of comparing via the computer a name for the source code with an identification of the data structure.

38. (Previously presented) The method of claim 34, further comprising having the computer perform the following steps:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

39. (Previously presented) The method of claim 38, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a configuration file.

40. (Previously presented) The method of claim 38, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a comment of the source code.

41. (Original) The method of claim 38, wherein the portion of the data structure comprises the first attribute field in the data structure.

42. (Original) The method of claim 34, wherein the source code comprises a class.

43. (Original) The method of claim 34, wherein the source code comprises a distributed computing component.

44. (Original) The method of claim 34, wherein the first attribute field in the data structure is related to a method in the source code.

45. (Currently amended) A method in a data processing system having a memory device with source code, the method comprising the steps of:

SSK/slc

providing a software development tool having a user interface that is operable by a user to automatically modify source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code;

determining via the computer whether the data structure is associated with the source code;

when a determination is made that the data structure is associated with the source code, storing the identification of the data structure in a comment of the source code to relate the data structure with the source code and another determination is made determining via the computer whether a first attribute field of the data structure is associated with a first attribute in the source code;

when it is determined that the first attribute field of the data structure is not associated with the first attribute of the source code, generating via the computer a source code modification, wherein the source code modification includes removing the first attribute field from the data structure;

determining via the computer whether a second attribute in the source code is associated with a second attribute field in the data structure;

when it is determined that the second attribute is not associated with the second attribute field in the data structure, generating via the computer another source code modification, wherein the another source code modification includes adding the second attribute field to the data structure;

receiving user input to modify at least a portion of [[modified]] the source code; converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification and the another source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in

the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

46. (Previously presented) The method of claim 45, further having the computer perform the following steps:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

47. (Previously presented) The method of claim 46, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a configuration file.

48. (Previously presented) The method of claim 46, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a comment of the source code.

49. (Original) The method of claim 46, wherein the portion of the data structure comprises the first attribute field in the data structure.

50. (Original) The method of claim 45, wherein the source code comprises a class.

51. (Original) The method of claim 45, wherein the source code comprises a distributed computing component.

52. (Currently amended) A method in a data processing system having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication that the source code has been modified; and automatically reflecting via the computer the modification in the data structure so as to avoid completely regenerating the data structure;

receive user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source

code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

53. (Previously presented) The method of claim 52, wherein the step of automatically reflecting the modification comprises having the computer perform the following steps:

determining whether a first attribute in the source code is associated with a first attribute field of the data structure; and when it is determined that a first attribute is not associated with a first attribute field in the data structure, generating the first attribute field in the data structure.

54. (Previously presented) The method of claim 52, further comprising having the computer perform the following steps: determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and when it is determined that a second attribute field is not associated with a second attribute in the source code, removing the second attribute field from the data structure.

55. (Previously presented) The method of claim 52, further having the computer perform the following steps:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

56. (Previously presented) The method of claim 55, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a configuration file.

57. (Previously presented) The method of claim 55, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a comment of the source code.

58. (Original) The method of claim 55, wherein the portion of the data structure comprises the first attribute field in the data structure.

59. (Original) The method of claim 52, wherein the source code comprises a class.

60. (Original) The method of claim 52, wherein the source code comprises a distributed computing component.

61. (Original) The method of claim 52, wherein the first attribute field in the data structure is related to a method in the source code.

62. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an identification of a data structure with an attribute field in a database of data structures useable to form an object-oriented element from the data structure; determining via the computer whether the data structure is associated with source code;

when it is determined that the data structure is associated with the source code, storing the identification of the data structure in a comment of the source code to relate the data structure

with the source code and determining via the computer whether the attribute field of the data structure is associated with an attribute in the source code;

when it is determined that the attribute field is not associated with an attribute in the source code, generating via the computer a source code modification, wherein the source code modification includes a new attribute in the source code [[from]] based upon the attribute field;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

63. (Previously presented) The computer-readable medium of claim 62, wherein the method further comprises having the computer perform the following steps:

when it is determined that the data structure is not associated with source code, retrieving a portion of the data structure; and generating the source code from the portion of the data structure.

64. (Previously presented) The computer-readable medium of claim 62, wherein the method further comprises having the computer perform the following steps:

when it is determined that the data structure is associated with source code, determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

when it is determined that a second attribute in the source code is not associated with a second attribute field in the data structure, removing the second attribute from the source code.

65. (Previously presented) The computer-readable medium of claim 64, wherein the step of removing the second attribute from the source code comprises the step of removing via the computer a method associated with the second attribute from the source code.

66. – 67. (Cancelled)

68. (Previously presented) The computer-readable medium of claim 65, wherein the method further comprises the step of modifying via the computer the graphical representation of the source code to reflect the removal of the second attribute.

69. (Previously presented) The computer-readable medium of claim 62, wherein the step of determining whether the data structure is associated with the source code comprises the step of searching via the computer a comment in the source code for the identification of the data structure.

70. (Previously presented) The computer-readable medium of claim 62, wherein the step of determining whether the data structure is associated with the source code comprises the

step of comparing via the computer a name for the source code with the identification of the data structure.

71. (Previously presented) The computer-readable medium of claim 62, wherein the method further comprises having the computer perform the following steps: retrieving access information for the database; and retrieving a portion of the data structure from the database using the access information.

72. (Previously presented) The computer-readable medium of claim 71, wherein the step of retrieving the access information comprises the step of retrieving via the computer the identification of the data structure and the access information from a configuration file.

73. (Previously presented) The computer-readable medium of claim 71, wherein the step of retrieving the access information comprises the step of retrieving via the computer the identification of the data structure and the access information from a comment of the source code.

74. (Original) The computer-readable medium of claim 71, wherein the portion of the data structure comprises the attribute field of the data structure.

75. (Original) The computer-readable medium of claim 62, wherein the source code comprises a class.

76. (Original) The computer-readable medium of claim 62, wherein the source code comprises a distributed computing component.

77. (Previously presented) The computer-readable medium of claim 76, wherein the distributed computing component is an Enterprise JavaBean.

78. (Previously presented) The computer-readable medium of claim 62, wherein the step of generating the new attribute in the source code comprises the step of generating via the computer a method in the source code to access the attribute field of the data structure.

79. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having source code that corresponds to a data structure within a database of data structures useable to form an object-oriented element from the data structure on a secondary storage device, the method comprising the steps of: providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool is for performing the following steps:

receiving an indication to update the source code;

determining via the computer whether the data structure is associated with the source code;

when a determination is made that the data structure is associated with the source code, storing the identification of the data structure in a comment of the source code to relate the data structure with the source code and another determination is made determining via the computer as to whether a first attribute in the source code is associated with a first attribute field in the data structure[[],];

when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, generating via the computer a source code modification,

wherein the source code modification includes removing via the computer the first attribute from the source code;

 determining via the computer whether a second attribute field in the data structure is associated with a second attribute in the source code;

 when it is determined that [[a]] the second attribute field is not associated with [[a]] the second attribute in the source code, generating via the computer another source code modification, wherein the another source code modification includes generating the second attribute in the source code [[from]] based upon the second attribute field and associating the second attribute with the second attribute field;

 receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification and the another source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

 modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

 using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

80. (Previously presented) The computer-readable medium of claim 79, wherein the

method further comprises the step of retrieving via the computer an identification of the data structure and access information for the secondary storage from a comment in the source code.

81. (Previously presented) The computer-readable medium of claim 80, wherein the method further comprises the step of retrieving via the computer a portion of the data structure from the secondary storage device using the access information.

82. (Original) The computer-readable medium of claim 81, wherein the portion comprises the first and the second attribute fields.

83. (Previously presented) The computer-readable medium of claim 79, wherein the step of removing the first attribute from the source code comprises the step of removing via the computer a method associated with the first attribute from the source code.

84. (Previously presented) The computer-readable medium of claim 79, wherein the step of generating the second attribute in the source code comprises the step of generating via the computer a method in the source code to access the second attribute field of the data structure.

85. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system including a computer to perform a method, the data processing system having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool is for performing the following steps: receiving an indication that the data source has been modified;

automatically reflecting via the computer the modification in the source code so as to avoid completely regenerating the source code;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

86. (Previously presented) The computer-readable medium of claim 85, wherein the step of automatically reflecting the modification comprises having the computer perform the following steps:

determining whether a first attribute in the source code is associated with a first attribute field in the data structure; and when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code.

87. (Previously presented) The computer-readable medium of claim 86, wherein the

step of removing the first attribute from the source code comprises the step of removing via the computer a first method associated with the first attribute in the source code.

88. (Previously presented) The computer-readable medium of claim 86, wherein the step of automatically reflecting the modification further comprises having the computer perform the following steps:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field in the data structure is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field.

89. (Previously presented) The computer-readable medium of claim 88, wherein the step of generating the second attribute in the source code comprises the step of generating via the computer a second method in the source code to access the second attribute field of the data structure.

90.-91. (Cancelled)

92. (Original) The computer-readable medium of claim 85, wherein the source code comprises a class.

93. (Original) The computer-readable medium of claim 85, wherein the source code comprises a distributed computing component.

94. (Previously presented) The computer-readable medium of claim 93, wherein the distributed computing component is an Enterprise JavaBean.

95. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system including a computer to perform a method, the data processing system having a memory device with source code, the method comprising the steps of: providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

determining via the computer whether the source code is associated with a data structure within a database of data structures useable to form and object-oriented element from the data structure;

when it is determined that the source code is associated with the data structure, storing an identification of the data structure in a comment of the source code to relate the data structure with the source code and determining via the computer whether a first attribute in the source code is associated with a first attribute field of the data structure;

when it is determined that the first attribute in the source code is not associated with the first attribute field in the data structure, generating the first attribute field in the data structure;

determining via the computer whether a second attribute field in the data structure is associated with a second attribute in the source code;

when it is determined that the second attribute field is not associated with the second attribute in the source code, removing via the computer the second attribute field from the data structure;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to

textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

96. (Previously presented) The computer-readable medium of claim 95, wherein the method further comprises the step of when it is determined that the source code is not associated with the data structure, generating via the computer the data structure from the source code.

97. (Previously presented) The computer-readable medium of claim 95, wherein the step of determining whether the source code is associated with the data structure comprises the step of searching via the computer a comment in the source code for an identification of the data structure.

98. (Previously presented) The computer-readable medium of claim 95, wherein the step of determining whether the source code is associated with the data structure comprises the step of comparing via the computer a name for the source code with an identification of the data structure.

99. (Previously presented) The computer-readable medium of claim 95, wherein the method further comprises having the computer perform the following steps:

SSK/slc

retrieving access information for a database that stores the data structure; and
retrieving a portion of the data structure from the database using the access information.

100. (Previously presented) The computer-readable medium of claim 99, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a configuration file.

101. (Previously presented) The computer-readable medium of claim 99, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a comment of the source code.

102. (Original) The computer-readable medium of claim 99, wherein the portion of the data structure comprises the first attribute field in the data structure.

103. (Original) The computer-readable medium of claim 95, wherein the source code comprises a class.

104. (Original) The computer-readable medium of claim 95, wherein the source code comprises a distributed computing component.

105. (Original) The computer-readable medium of claim 95, wherein the first attribute field in the data structure is related to a method in the source code.

106. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a memory device with source code, the method comprising the steps of:

SSK/slc

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code;

determining via the computer whether the data structure is associated with the source code;

when a determination is made that the data structure is associated with the source code, storing an identification of the data structure in a comment of the source code to relate the data structure with the source code and another determination is made determining via the computer whether a first attribute field of the data structure is associated with a first attribute in the source code;

when it is determined that the first attribute field of the data structure is not associated with the first attribute of the source code, removing via the computer the first attribute field from the data structure;

determining via the computer whether a second attribute in the source code is associated with a second attribute field in the data structure;

when it is determined that the second attribute is not associated with the second attribute field in the data structure, adding the second attribute field to the data structure;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

107. (Previously presented) The computer-readable medium of claim 106, wherein the method further comprises having the computer perform the following steps:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

108. (Previously presented) The computer-readable medium of claim 107, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a configuration file.

109. (Previously presented) The computer-readable medium of claim 107, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a comment of the source code.

110. (Original) The computer-readable medium of claim 107, wherein the portion of the data structure comprises the first attribute field in the data structure.

111. (Original) The computer-readable medium of claim 106, wherein the source code comprises a class.

112. (Original) The computer-readable medium of claim 106, wherein the source code comprises a distributed computing component.

113. (Currently Amended) A computer-readable medium containing instructions for controlling a data processing system including a computer to perform a method, the data processing system having a memory device with source code and a secondary storage device with a data structure corresponding to the source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication that the source code has been modified; and automatically reflecting via computer the modification in the data structure so as to avoid completely regenerating the data structure;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source

code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

114. (Previously presented) The computer-readable medium of claim 113, wherein the step of automatically reflecting the modification comprises having the computer perform the following steps:

determining whether a first attribute in the source code is associated with a first attribute field of the data structure; and

when it is determined that a first attribute is not associated with a first attribute field in the data structure, generating the first attribute field in the data structure.

115. (Previously presented) The computer-readable medium of claim 113, wherein the method further comprises having the computer perform the following steps:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field is not associated with a second attribute in the source code, removing the second attribute field from the data structure.

116. (Previously presented) The computer-readable medium of claim 113, wherein the method further comprises having the computer perform the following steps:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

117. (Previously presented) The computer-readable medium of claim 116, wherein

SSK/slC

the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a configuration file.

118. (Previously presented) The computer-readable medium of claim 116, wherein the step of retrieving the access information comprises the step of retrieving via the computer an identification of the data structure and the access information from a comment of the source code.

119. (Original) The computer-readable medium of claim 116, wherein the portion of the data structure comprises the first attribute field in the data structure.

120. (Original) The computer-readable medium of claim 113, wherein the source code comprises a class.

121. (Original) The computer-readable medium of claim 113, wherein the source code comprises a distributed computing component.

122. (Original) The computer-readable medium of claim 113, wherein the first attribute field in the data structure is related to a method in the source code.

123. (Currently amended) A data processing system comprising:
a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code;

a secondary storage device further comprising source code that corresponds to a data structure within a database of data structures useable to form an object-oriented element from the data structure on a secondary storage device;

SSK/slc

a computer including a memory device further comprising a program that receives an indication to update the source code, ~~that determines whether a first attribute in the source code is associated with a first attribute field in the data structure;~~

determining via the computer whether the data structure is associated with the source code;

when a determination is made that the data structure is associated with the source code, storing an identification of the data structure in a comment of the source code to relate the data structure with the source code and another determination is made as to whether a first attribute in the source code is associated with a first attribute field in the data structure;

when determined that a first attribute in the source code is not associated with a first attribute field in the data structure, the program generates a source code modification wherein the source code modification includes removing ~~removes~~ the first attribute from the source code, ~~the program further determines;~~

determining whether a second attribute field in the data structure is associated with a second attribute in the source code;

when determined that a second attribute field is not associated with a second attribute in the source code, the computer program generates another source code modification wherein the source code modification includes generating the second attribute in the source code from the second attribute field and associating via computer the second attribute with the second attribute field;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification and the another source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

124. (Previously presented) The data processing system of claim 123, wherein the computer program further retrieves via the computer an identification of the data structure and access information for the secondary storage from a comment in the source code.

125. (Previously presented) The data processing system of claim 124, wherein the computer program further retrieves a portion of the data structure from the secondary storage device using the access information.

126. (Original) The data processing system of claim 125, wherein the portion comprises the first and the second attribute fields.

127. (Previously presented) The data processing system of claim 123, wherein when the computer program removes the first attribute from the source code, the computer program removes a method associated with the first attribute from the source code.

128. (Previously presented) The data processing system of claim 123, wherein when the computer program generates the second attribute in the source code, the computer program generates a method in the source code to access the second attribute field of the data structure.

129. (Currently amended) A data processing system comprising:

a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code; a secondary storage device further comprising source code;

a computer including a memory device further comprising a computer program that receives an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code, that determines whether a first attribute field of the data structure is associated with a first attribute in the source code;

when determined that the data structure is associated with the source code, the computer program stores an identification of the data structure in a comment of the source code to relate the data structure with the source code and

when determined that the first attribute field of the data structure is not associated with the first attribute of the source code, the computer program removes the first attribute field from the data structure, the computer program further determines whether a second attribute in the source code is associated with a second attribute field in the data structure;

when determined that the second attribute is not associated with the second attribute field in the data structure, the computer program adds the second attribute field to the data structure;

receiving user input to modify the source code;

converting the source code modification to generating a transient meta model after receiving user input, wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual

representation of at least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

130. (Previously presented) The data processing system of claim 129, wherein when the program further retrieves access information for a database that stores the data structure, and retrieves a portion of the data structure from the database using the access information.

131. (Previously presented) The data processing system of claim 130, wherein when the computer program retrieves the access information, and the computer program retrieves an identification of the data structure and the access information from a configuration file.

132. (Previously presented) The data processing system of claim 130, wherein when the computer program retrieves the access information, the computer program retrieves an identification of the data structure and the access information from a comment of the source code.

133. (Original) The data processing system of claim 130, wherein the portion of the data structure comprises the first attribute field in the data structure.

134. (Original) The data processing system of claim 129, wherein the source code comprises a class.

135. (Original) The data processing system of claim 129, wherein the source code comprises a distributed computing component.

136. (Currently Amended) A system including a computer having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the system comprising:

a software development tool having a user interface that is operable by a user to modify source code;

wherein the software development tool includes computer instructions for storing the identification of the data structure in a comment of the source code to relate the data structure with the source code when a determination is made that the data structure is associated with the source code;

wherein the software development tool includes computer instructions for receiving an indication that the data structure has been modified; and computer instruction for automatically reflecting the modification in the source code so as to avoid completely regenerating the source code;

wherein the software development tool includes computer instructions for converting the source code modification to generating a transient meta model after reflecting the modification in the source code ,wherein the transient meta model includes a language-neutral representation of the source code modification, and wherein the language-neutral representation is configured to textually display the source code modification in the language and configured to display a graphical representation of at least a portion of the database of data structures;

wherein the software development tool includes computer instruction for modifying and displaying via the computer a graphical representation of at least a portion of the language-neutral representation to reflect the source code modification and a textual representation of at

least a portion of the language-neutral representation to reflect the source code modification, wherein the textual representation and the graphical representation are displayed simultaneously; and

a QA module configured to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.